

```
In [2]: # Download The Data From Kaggle Website
# Ref --->>> https://www.kaggle.com/c/m5-forecasting-accuracy/data

#!wget --header="Host: storage.googleapis.com" --header="User-Agent: Mozilla/5.0
```

```
In [4]: # Unzip the data

#!unzip m5-forecasting-accuracy.zip
```

```
In [91]: # import the Library

import pandas as pd
import numpy as np
```

```
In [92]: # Read the Calander Data

cal = pd.read_csv('calendar.csv')
```

```
In [107]: cal.tail()
```

```
Out[107]:
```

	date	wm_yr_wk	weekday	wday	month	year	d	event_name_1	event_type_1	ev
1964	2016-06-15	11620	Wednesday	5	6	2016	d_1965	NaN	NaN	
1965	2016-06-16	11620	Thursday	6	6	2016	d_1966	NaN	NaN	
1966	2016-06-17	11620	Friday	7	6	2016	d_1967	NaN	NaN	
1967	2016-06-18	11621	Saturday	1	6	2016	d_1968	NaN	NaN	
1968	2016-06-19	11621	Sunday	2	6	2016	d_1969	NBAFinalsEnd	Sporting	

```
In [94]: cal.shape
```

```
Out[94]: (1969, 14)
```

```
In [95]: # You can Remove Also, Later Part i didn't Use

x1 = cal['event_name_1'].copy()
x2 = cal['event_name_2'].copy()
```

Task-1. I only Take Festival which Falls into Cross Validation And Test Data.

In [96]: *# ALL Days in Cross Validation And Test Data*

```
event_day = ['d_{}'.format(c) for c in list(np.arange(1910,1970))]  
event_day[-5:]
```

Out[96]: ['d_1965', 'd_1966', 'd_1967', 'd_1968', 'd_1969']

In [97]: *# Event1 in event_day*

```
event1_fest = cal[cal['d'].isin(event_day)]['event_name_1'].unique().tolist()  
event1_fest
```

Out[97]: [nan,
 'Pesach End',
 'OrthodoxEaster',
 'Cinco De Mayo',
 "Mother's day",
 'MemorialDay',
 'NBAFinalsStart',
 'Ramadan starts',
 'NBAFinalsEnd']

In [98]: *# Event2 in event_day*

```
event2_fest = cal[cal['d'].isin(event_day)]['event_name_2'].unique().tolist()  
event2_fest
```

Out[98]: [nan, "Father's day"]

In [99]: *# Combine Both Events*

```
events = event1_fest+event2_fest  
events
```

Out[99]: [nan,
 'Pesach End',
 'OrthodoxEaster',
 'Cinco De Mayo',
 "Mother's day",
 'MemorialDay',
 'NBAFinalsStart',
 'Ramadan starts',
 'NBAFinalsEnd',
 nan,
 "Father's day"]

In [100]: *# Do Some Formatting*

```
aal_fes = ['event_name_1_{}'.format(c) for c in events]+['event_name_2_{}'.format(c) for c in events]
aal_fes[0:5]
```

Out[100]: ['event_name_1_nan',
 'event_name_1_Pesach End',
 'event_name_1_OrthodoxEaster',
 'event_name_1_Cinco De Mayo',
 "event_name_1_Mother's day"]

In [101]: *# ref --->> <https://stackoverflow.com/questions/36631163/what-are-the-pros-and-cons-of-one-hot-encoding>*

#A. OneHotEncoder cannot process string values directly. If your nominal features are strings, you need to convert them to integers first.
#B. pandas.get_dummies is kind of the opposite. By default, it only converts string values to dummy variables.

```
calendar = pd.concat([cal, pd.get_dummies(cal[['event_name_1', 'event_name_2'])]).drop(['event_name_1', 'event_name_2'], axis=1)
calendar.head()
```

Out[101]:

	date	wm_yr_wk	weekday	wday	month	year	d	event_type_1	event_type_2	snap_CA
0	2011-01-29	11101	Saturday	1	1	2011	d_1	NaN	NaN	0
1	2011-01-30	11101	Sunday	2	1	2011	d_2	NaN	NaN	0
2	2011-01-31	11101	Monday	3	1	2011	d_3	NaN	NaN	0
3	2011-02-01	11101	Tuesday	4	2	2011	d_4	NaN	NaN	1
4	2011-02-02	11101	Wednesday	5	2	2011	d_5	NaN	NaN	1

In [102]: *# Events in Cross Validation And Test DataSet*

```
events_Present_in_df = [c for c in calendar.columns if c in aal_fes]
events_Present_in_df
```

Out[102]: ['event_name_1_Cinco De Mayo',
"event_name_1_Father's day",
'event_name_1_MemorialDay',
"event_name_1_Mother's day",
'event_name_1_NBAFinalsEnd',
'event_name_1_NBAFinalsStart',
'event_name_1_OrthodoxEaster',
'event_name_1_Pesach End',
'event_name_1_Ramadan starts',
'event_name_2_Cinco De Mayo',
"event_name_2_Father's day",
'event_name_2_OrthodoxEaster']

In [103]: calendar.columns

Out[103]: Index(['date', 'wm_yr_wk', 'weekday', 'wday', 'month', 'year', 'd',
'event_type_1', 'event_type_2', 'snap_CA', 'snap_TX', 'snap_WI',
'event_name_1_Chanukah End', 'event_name_1_Christmas',
'event_name_1_Cinco De Mayo', 'event_name_1_ColumbusDay',
'event_name_1_Easter', 'event_name_1_Eid al-Fitr',
'event_name_1_EidAlAdha', 'event_name_1_Father's day',
'event_name_1_Halloween', 'event_name_1_IndependenceDay',
'event_name_1_LaborDay', 'event_name_1_LentStart',
'event_name_1_LentWeek2', 'event_name_1_MartinLutherKingDay',
'event_name_1_MemorialDay', 'event_name_1_Mother's day',
'event_name_1_NBAFinalsEnd', 'event_name_1_NBAFinalsStart',
'event_name_1_NewYear', 'event_name_1_OrthodoxChristmas',
'event_name_1_OrthodoxEaster', 'event_name_1_Pesach End',
'event_name_1_PresidentsDay', 'event_name_1_Purim End',
'event_name_1_Ramadan starts', 'event_name_1_StPatricksDay',
'event_name_1_SuperBowl', 'event_name_1_Thanksgiving',
'event_name_1_ValentinesDay', 'event_name_1_VeteransDay',
'event_name_2_Cinco De Mayo', 'event_name_2_Easter',
'event_name_2_Father's day', 'event_name_2_OrthodoxEaster'],
dtype='object')

In [104]: calendar = calendar[['date', 'wm_yr_wk', 'weekday', 'wday', 'month', 'year', 'd',
'event_type_1', 'event_type_2',
'snap_CA', 'snap_TX', 'snap_WI'] + events_Present_in_df]

Task-2. Give Weight to Each Festivals.

```
In [105]: # Give Last 30 Days Festivle Importance

for col in [c for c in calendar.columns.tolist() if 'event_name' in c]:
    days_event = np.where(calendar[col] == 1)[0].tolist()
    calendar[col] = calendar['d']

    dict_days_event = {}
    for d in days_event:
        for i in range(0, 30):
            dict_days_event['d_'+str(d-i)] = 30-i

    calendar[col] = calendar[col].map(dict_days_event).fillna(0)
```

```
In [106]: calendar.tail()
```

```
Out[106]:
```

	date	wm_yr_wk	weekday	wday	month	year	d	event_type_1	event_type_2	sna
1964	2016-06-15	11620	Wednesday	5	6	2016	d_1965	NaN	NaN	
1965	2016-06-16	11620	Thursday	6	6	2016	d_1966	NaN	NaN	
1966	2016-06-17	11620	Friday	7	6	2016	d_1967	NaN	NaN	
1967	2016-06-18	11621	Saturday	1	6	2016	d_1968	NaN	NaN	
1968	2016-06-19	11621	Sunday	2	6	2016	d_1969	Sporting	Cultural	

```
In [108]: calendar['Event_1'] = x1
calendar['Event_2'] = x2
```

```
In [110]: calendar.tail()
```

```
Out[110]:
```

	date	wm_yr_wk	weekday	wday	month	year	d	event_type_1	event_type_2	sna
1964	2016-06-15	11620	Wednesday	5	6	2016	d_1965	NaN	NaN	
1965	2016-06-16	11620	Thursday	6	6	2016	d_1966	NaN	NaN	
1966	2016-06-17	11620	Friday	7	6	2016	d_1967	NaN	NaN	
1967	2016-06-18	11621	Saturday	1	6	2016	d_1968	NaN	NaN	
1968	2016-06-19	11621	Sunday	2	6	2016	d_1969	Sporting	Cultural	

```
In [111]: # Save ALL The Data in Google Drive

from google.colab import drive
drive.mount('/drive')
calendar.to_pickle('/drive/My Drive/Case_Study1/Calander_Data_Preprocessing_2.pkl')
```

Drive already mounted at /drive; to attempt to forcibly remount, call drive.mount("/drive", force_remount=True).